

## **AMENDMENTS TO THE CLAIMS**

This listing of claims replaces all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (Currently Amended) A client system that facilitates source code control, comprising:

a processor; and

one or more physical computer readable storage media operatively coupled to the processor, the computer-readable storage media having stored thereon computer executable instructions that, when executed by the processor, are configured to implement the client system, including:

a client-side client workspace that stores at least one source code file downloaded from a server-side source control repository that maintains state information for one or more source code files stored in the repository, the client workspace enabling a user of the client system to checkout the source code file for modification by the user and to store the source code file when modified;

a client-side client source code control component that tracks an activity associated with a modification of the source code file in the client workspace when the client is in an offline mode, and that transmits the activity during an update process when the client moves to an online mode, the client source code control component comprising:

a file cache that stores a pristine copy of the source code file in an unmodified state in response to the source code file being checked out for modification from the client-side client workspace; and

an activity list that stores the activity, the activity comprising one or more commands executed against the source code file during the offline mode and associated with the modification; and

wherein the client-side client source code control component is configured to interface with a server-side server source code control component to facilitate transfer of the activity and update of the source code file to the server-side source control repository, ~~wherein one of the client-side client source code control component or the server-side server source code control component checks for an error during the update~~

~~process and determines whether the update can proceed or must be aborted in part or whole.~~

2. (Canceled)
3. (Previously Presented) The client system of claim 1, wherein the source code file stored in the file cache remains in an unmodified state.
4. (Previously Presented) The client system of claim 1, wherein contents of the file cache are maintained in both the offline mode and online mode of the client.
5. (Previously Presented) The client system of claim 1, wherein the client workspace stores all source code files that have been at least one of modified and deleted.
6. (Previously Presented) The client system of claim 1, wherein the client downloads the source code file from the server-side source control repository before the client moves to the offline mode.
7. (Canceled)
8. (Previously Presented) The client system of claim 1, wherein the file cache also stores at least one of pending changeset data, a filetype definition, and a site-specific help file.
9. (Previously Presented) The client system of claim 1, wherein an error is resolved during a reconciliation process of the activity to the source code file before the source code file can be updated with the modification.
10. (Canceled)
11. (Previously Presented) The client system of claim 1, wherein the source code file is downloaded into the client workspace before the client moves to the offline mode.

12. (Previously Presented) The client system of claim 1, wherein a pristine copy of the source code file is automatically loaded into a client file cache in response to a checkout-related command being executed.

13. (Previously Presented) The client system of claim 1, wherein the activity list stores the activity in an arbitrary order which need not be sequential, such that the client checks the activity list for an error during the update process and determines whether to transmit the activity list when the client moves to the online mode or to abort a part or all of the update process.

14. (Previously Presented) The client system of claim 1, wherein the source code file stored in the file cache remains in an unmodified state, and is a file that has been at least one of modified and deleted in the client workspace.

15. (Previously Presented) The client system of claim 1, wherein the source code file is stored in the file cache of the client during a checkout process that is executed when the client is in one of the offline mode and the online mode.

16. (Canceled)

17. (Previously Presented) The client system of claim 1, wherein the activity is persisted to a server to update a server source code file associated with the source code file during an update process associated with the online mode.

18-19. (Canceled)

20. (Previously Presented) The client system of claim 1, further comprising a classifier that automates a source code control feature by making an inference based on data associated with at least one of the online mode and the offline mode.

21. (Previously Presented) In a network comprising a client system that facilitates course code control, a computer program product for implementing a method of tracking and reconciling offline file editing and enforcing file security, the computer program product comprising one or more physical computer-readable storage media having stored thereon computer-executable instructions that, when executed by a processor, cause the client system to perform the following::

establishing a client workspace to store a plurality of source code files, the client workspace established to allow a user of the client to checkout a source code file for modification by the user and to store the source code file when modified;

downloading a copy of a remote source code file from a server source code control repository to the client workspace;

moving the client system to an offline mode;

executing one or more commands to checkout the source code file in the client workspace for editing by the user, and in response to the checkout commands, caching a pristine copy of the source code file in a client file cache, wherein the pristine copy is usable in the offline mode to facilitate undo and difference processes;

modifying the source code file in the client workspace by executing one or more commands against the source code file;

storing activity data in an activity list, which activity data includes the one or more commands executed against the source code file;

moving the client system to an online mode;

performing an error check to determine if a security error exists, including:

determining if the remote source code file is locked; and

determining if an administrator has disallowed an update process;

reconciling the activity data with the remote source code file by transmitting the activity data to the server to update the remote source code file if no security error is detected;

resolving any conflicts that occur during the reconciliation process; and

uploading the modified source code file to the server source code control repository when any conflicts have been resolved.

22-27. (Cancelled)

28. (Previously Presented) At a client system that facilitates source code control, a method of tracking and reconciling offline file editing and enforcing file security, the method comprising acts of:

establishing a client workspace to store a plurality of source code files, the client workspace established to allow a user of the client to checkout a source code file for modification by the user and to store the source code file when modified;

downloading a copy of a remote source code file from a server source code control repository to the client workspace;

moving the client system to an offline mode;

executing one or more commands to checkout the source code file in the client workspace for editing by the user, and in response to the checkout commands, caching a pristine copy of the source code file in a client file cache, wherein the pristine copy is usable in the offline mode to facilitate undo and difference processes;

modifying the source code file in the client workspace by executing one or more commands against the source code file;

storing activity data in an activity list, which activity data includes the one or more commands executed against the source code file;

moving the client system to an online mode;

performing an error check to determine if a security error exists, including:

determining if the remote source code file is locked; and

determining if an administrator has disallowed an update process;

reconciling the activity data with the remote source code file by transmitting the activity data to the server to update the remote source code file if no security error is detected;

resolving any conflicts that occur during the reconciliation process; and

uploading the modified source code file to the server source code control repository when any conflicts have been resolved.

29. (Original) The method of claim 28, further comprising storing information at the client before entering the offline mode.

30. (Original) The method of claim 28, further comprising updating a checkout record at the server during the online mode.

31. (Original) The method of claim 28, further comprising issuing a command to enter the offline mode, and a corresponding command to enter the online mode.

32. (Original) The method of claim 28, further comprising issuing a command that includes a URL to a workspace.

33. (Previously Presented) The method of claim 28, further comprising:  
checking out a source code file during the act of downloading, which is during the online mode.

34. (Original) The method of claim 28, further comprising:  
detecting an error during the offline mode;  
presenting an error message associated with the error;  
maintaining the client in the offline mode in response to detecting the error; and allowing the client to move to the online mode after the error has been resolved.

35. (Original) The method of claim 28, further comprising imposing permissions required for the offline mode, during the online mode.

36. (Original) The method of claim 28, further comprising caching at the client at least one of unmodified files, pending change set information, filetype definitions, and site-specific help files.

37. (Previously Presented) The method of claim 28, further comprising reapplying a checkout process to the server when at least one of a checkout was cancelled at the server when

the client was offline and a checkout was performed offline on the client after the source code file was downloaded to the client without the checkout process issued to the server during the online mode.

38-41. (Canceled)